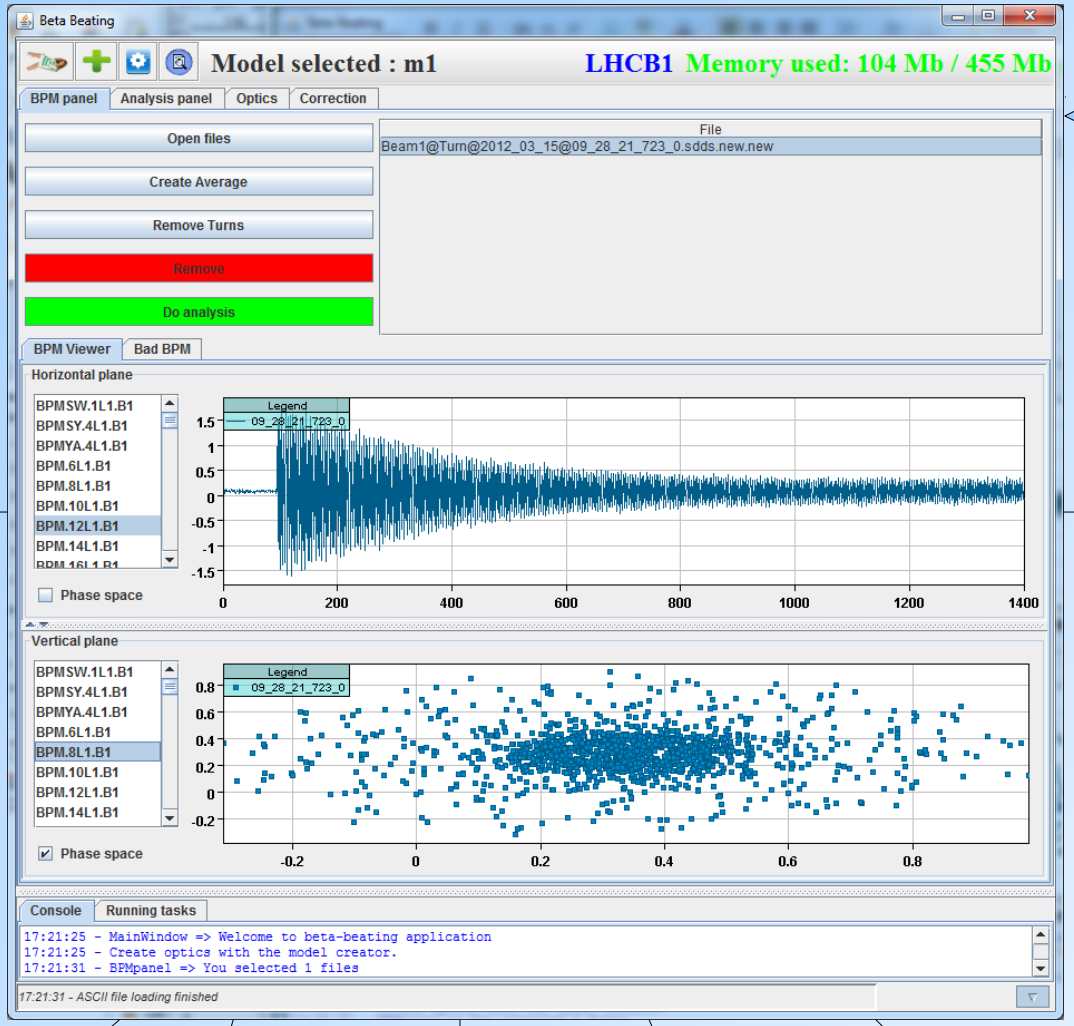
OMC Software Improvements



LSA

MAD-X

C/FORTRAN program

Multiple Python scripts

- Achievements for the last 20 months
- Current situation
- Goals

Achievements

- Computer scientist joined the team in 2012-02
- Refactor the Java software used at the CCC
- Define a release process
- Improve source code in terms of code quality, performance and features

Examples for improvements

- Reliability

For the last 10 months during the LHC runs, there were no severe problems with the Java GUI anymore which could have prevented the OMC team from working at the CCC

Reliability – How?

- Better source code quality
- Unit and regression tests
- Release procedure before every new GUI version

Examples for improvements

- Efficiency

Improved memory handling and running time for Java GUI

Efficiency

Name	Before	After
GUI startup	45s	<1s
Model creation	10 - 15min	1min
File loading	1min	5s
SVD clean	2 - 5min	6 - 10s
DRIVE	1min - 20min	10s
Rest	30s	30s
Approximated total	25min	2min
Memory usage per file	50 - 100MB	12MB
Maximum files	Around 10	> 50

Maintainability

- Hard to measure
- Software architecture and design patterns
- Remove unnecessary/dead code (we have SVN/GIT)
- Code conventions
- Documentation (Speaking names instead of cryptic names with comments)
- Separation of concerns

Example – Clean code

```
/* mean from double[] */
public double f(double[] d){
    double x=0.0;
    for(int i=0;i<d.length;i++)
       x=x+d[i];
    return x;
}
```

Example – Clean code

```
/* mean from double[] */
public double f(double[] d){
    double x=0.0;
    for(int i=0;i<d.length;i++)
       x=x+d[i];
    return x;
}
```

```
/** Standard arithmetic mean: 1/n * sum from i=1 to n over x_i */
public static double arithmeticMean(final double... values) {
    double sum = 0.0d;
    for (final double valuesItem : values)
        sum += valuesItem;
    return sum / values.length;
}
```

Current Situation

- LS1 → time for major changes until LS1 is finished
- Small team → Fast and rapid development process
- Software in C / FORTRAN77 / Java / Python → Hard to be expert in everything
- Probably „permanent“ support from computer scientists → Working together as experts for each domain

Goals

- Working software after LS1
- Stable environment for fast/agile development
- Fast software usage without bigger delays
- Improve Python source codes
- Documentation for long term support and possible usage outside CERN/LHC

Goals

- Working software after LS1
- Stable environment for fast/agile development
- Fast software usage without bigger delays
- Improve Python source codes
- Documentation for long term support and possible usage outside CERN/LHC

Software should support, not create more work.

How to reach the goals?

goto VIKTOR;